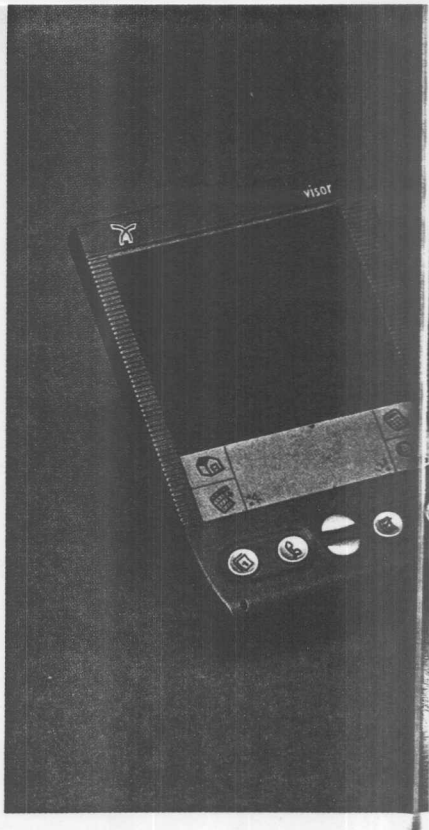


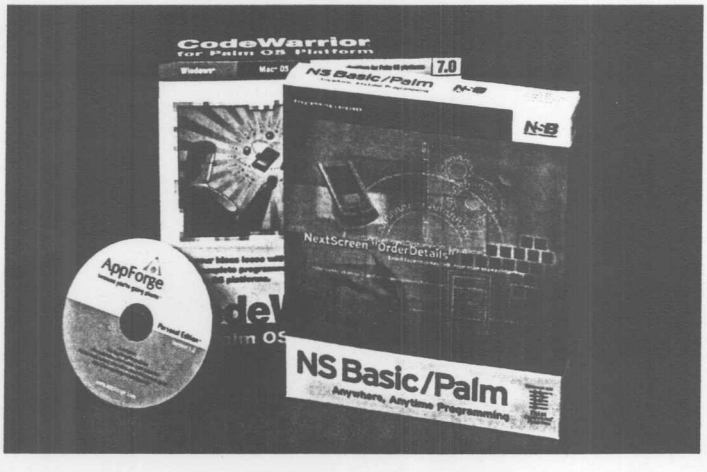
Jon Titus, *Editorial Director*

Put an instrument in your PALM

Software tools put custom programs in a PDA.

By now, you've probably had a close look at a Palm Pilot, Handspring Visor, Sony Clié, or one of the other portable digital assistants (PDAs) that use the Palm operating system (Palm OS). I bet you've wondered how difficult it would be to develop your own software that would run on a Palm-like PDA. Palm OS device suppliers and many third parties offer an abundance of tools and a wealth of assistance. You just need to know where to look.

Palm Computing, a division of 3Com Corp. (Santa Clara, CA), developed the family of Palm Pilot PDAs and now licenses the Palm OS to other suppliers including Handspring, Sony, Symbol Technologies, Nokia, and Samsung (**Figure 1**). Each licensee can use the Palm OS in its products, and in general, a program developed to run under the Palm OS on one manufacturer's device should run on Palm OS devices from other manufacturers. As with most operating systems, several versions of the Palm OS exist.



At its most basic level, programming a Palm OS device involves writing code that will run on a Motorola DragonBall (MC68328) processor. But don't break into a cold sweat thinking about programming in assembly language. Third parties offer products that can ease your entry into programming Palm OS devices at a high level. These products include C and C++ compilers, emulators, a Visual Basic add-in, and a stand-alone Basic compiler. Books, application notes, user Web sites, and even an online Web university can point you in the right direction. (See, "For more information," p. 16.)

You should keep in mind that the Palm OS PDAs are nothing more than small embedded systems. Yes, you can do neat things with a PDA, but it's simply a battery-powered portable system, and it lacks the hardware and software capabilities of a computer that runs Windows or Mac OS. The developers of the Palm OS never envisioned the sorts of things people now use PDAs to do. You can "extend" the capabilities of a PDA, but don't expect one to operate like a desktop PC.

One limitation is that programs must run with small stacks, and jump instructions can only jump a maximum of 32 kbytes forward or backward. You can use "longer" jumps, but development systems make you go through many extra steps to do so. Dynamic heaps—used to store variables and other information—have limits, too, depending on the version of Palm OS you plan to use. So, you'll have to pay careful attention to the system resources your application will use.

The Palm OS isn't Windows

The Palm OS offers neither the number nor the variety of the application program interfaces (APIs) found in a PC's operating system. If you want to draw a graph on the small (160x160-pixel) screen, for example, you must figure out how to place axes, legends, and data points on the screen. The

Palm OS contains no graph-drawing API call. In short, you can't program a PDA at the same level you would a PC. Programmers may offer add-in shared libraries and special functions, though, and you can search for them on the Web.

So, where do you get started? If you're a C or C++ programmer, you might start with the freeware tools—collectively called the PRC-tools package—available from at least two sources. The PRC (Pilot Resource Code) tools—patched versions of the GNU software—include the GNU C Compiler (GCC), the GNU DeBugger (GDB), the binutils, and various linker tools. You won't get much help using these tools except through online discussion groups and newsgroups. The Source Forge Web site (sourceforge.net) provides several program examples that may prove helpful. Search the site for "Palm OS."

If you're not a C or C++ programmer—and don't want to become one—you can apply your knowledge of Basic to write application software that runs under the Palm OS. I'll get to some plain Basic and Visual Basic tools shortly.

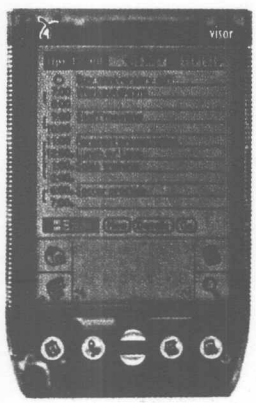


FIGURE 1 A PDA such as the Visor from Handspring uses the Palm OS to control its operations. The OS provides function calls and "hooks" that applications software can use to monitor inputs, put information on the screen, or operate I/O ports. *Courtesy of Handspring.*

Emulate your application

To test software, you'll need a copy of the free Palm OS emulator (POSE) that you can download from the Palm OS Web site (www.palmos.com). The emulator relies on an "image" of the Palm OS ROM, and to download it you must first join the Palm Alliance Program. Membership is free, but you must return a signed agreement before you can access the ROM code. As a member, you also can gain access to some of the source code for the ROM. The POSE comes with a utility program that lets you "extract" a ROM image from your own Palm OS-based device—I'll call it the target system—so you can ensure your code will work with a specific model or family of devices.

The POSE runs on your development PC and lets you test your software as if it were running under the Palm OS on your target. Thus, you don't have to transfer each new revision to the target and then try it. All software development takes place on your PC. When at last you have bug-free code—OK, almost bug-free code—you can load your program into the target and try it. You use the standard synchronization tools provided with the PDA to load the compiled code, which is saved in a .prc file, to the target just as you would any other program, such as a game.

Free tools have a place, but you may decide to invest in commercial tools. If you want to continue using C or C++, the CodeWarrior 7.0 package ($369) from Metrowerks (Austin, TX) may answer your needs. (See, "Commercial sources," p. 14.) This software provides a complete set of development tools you can use with either a Windows-based or a Mac OS-based PC. The CodeWarrior's integrated development environment (IDE) came from the Macintosh world, so Windows programmers may need a bit of time to get used to it (**Figure 2**). The CodeWarrior software compiles code that runs directly under the Palm OS.

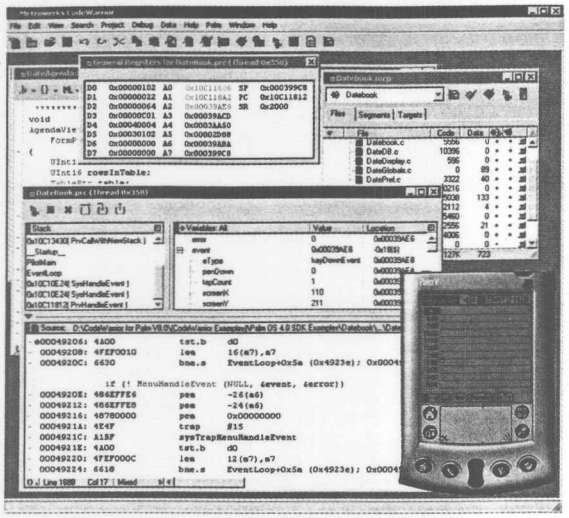


FIGURE 2 The Metrowerks CodeWarrior C/C++ development tools let you write and compile programs on a PC host. *Courtesy of Metrowerks.*

If you want to evaluate the CodeWarrior tool, you can download a demo version (45 Mbytes) of the software from the Metrowerks Web site. Or, you can request a free CD-ROM with the demo software on it. The CD-ROM lacks some documentation files, but you can download them from the Web site. The demo software limits the linker output to 4 kbytes, and it limits the final executable .prc file to 13.5 kbytes. The Metrowerks site provides a series of free self-paced tutorials on how to use the CodeWarrior tools to develop software for Palm OS devices.

Visor offers an expansion slot

The PDA you choose to host application code will depend on the capabilities you need. If you require a simple data-collection PDA, a basic Palm or Handspring Visor should handle your needs. If you want to add specialized hardware to a PDA, the Handspring Visor may be your best bet. Other PDAs offer various slots for add-in modules, but none seem as flexible as that in the Handspring product line (**Figure 3**).

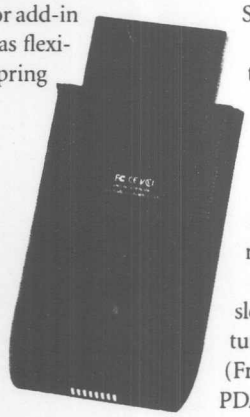


FIGURE 3 Many PDAs offer some sort of expansion capability. The Visor includes a Springboard slot that supplies address, data, and control lines.
Courtesy of Handspring.

The Visor family comes with a Springboard slot that lets you develop drop-in hardware that can extend the capabilities of a basic unit. Consumers can now choose from a variety of modules, including a digital camera, a cell phone, an MP3 player, and a language translator. You can buy a few instrumentation modules, too (**Figure 4**).

You can design Springboard modules, but you don't have to do all the development work. The Handspring Web site (www.handspring.com) lists several developers who can design products, write code, or deliver a complete product. Look under "resources & partners" in the Developers section. Even if you plan only a small hardware project, you might want to purchase a Handspring Visor Diagnostic Card from efig.com. This module includes a UART based on the popular 16C650 chip, 2 Mbytes of FLASH memory, a pushbutton, two LEDs, and an internal 16-bit register you can read from and write to. The diagnostic card provides simple circuitry and software that let you experiment. You can use all the devices in the module as if they exist within the Visor itself.

When you plug a module into the Visor's Springboard slot, the Visor senses the module's presence and "installs" links to the module's code in a shared library that application programs can access. When you remove a module, the Palm OS recognizes that action and removes the shared-library references for the module. Thus, you can add or remove a module at any time.

Not all modules require an expansion slot, though. Some, such as the temperature-measurement add-on from Tescina (Fremont, CA), use the serial port on a PDA (**Figure 5**).

What if C is not for me?

Not everyone wants to program Palm OS applications using C or C++. Two commercial software products—AppForge and NS Basic/Palm—let programmers take advantage of their experience using Microsoft's Visual Basic 6.0 (VB). And the Basic tools will appeal to engineers who want to get a quick start before they jump into using a C/C++ development system that can come with a steep learning curve.

The AppForge software provides 18 special functions called "Ingots" that supply radio buttons, list boxes, tones, serial communications, and similar functions. You drag and drop the controls and then set their attributes through standard property boxes as you would in VB. And you write code just as if you were writing an application using VB. Because you compile application code so it runs on a host PC, you can test code without leaving the VB programming environment. When you're ready to try your software on a Palm OS target, simply transfer the program to the PDA and run it. Applications run under Palm OS 3.1 or higher.

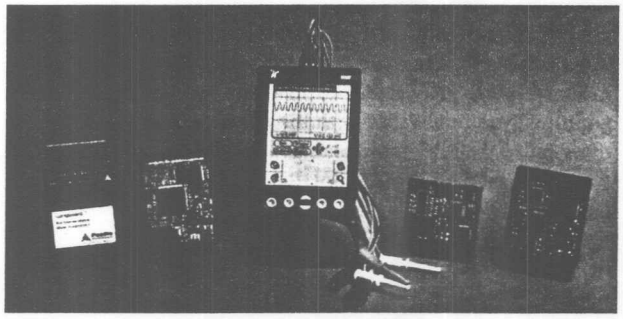


FIGURE 4 The Springboard slot in a Visor can accept many types of add-in modules, such as those shown here. *Courtesy of Pacific Microinstruments.*

Applications written with AppForge rely on a virtual machine (or run-time module)—the AppForge Booster—within your target. The Booster requires about 300 kbytes of memory space, and it works with a specific microprocessor and operating system. If you write an application and decide to move it to another target, say a Pocket PC, for which AppForge supplies a Booster, your original application should run without modification. AppForge provides the Booster for free, so you can distribute your applications unencumbered by a license fee or royalty charge.

Because AppForge relies on a virtual machine in the Palm-OS based device, it can be

too slow for applications that require a lot of processor time. If you use AppForge to write code that contains an FFT routine, for example, the FFT's numerous math operations may slow the application considerably. But if you need to do a few calculations and display the results, AppForge should do the trick.

AppForge 2.0 comes in either a personal edition ($69) or a development package ($699). The development package provides features that include ODBC and support for wireless devices. The next version of the product, due for release as we go to press, will provide the capability to link AppForge applications with functions written in C/C++, and users will be able to compile applications for the Palm OS or for the Pocket PC OS. AppForge also will offer a software-development kit (SDK) that will allow developers to integrate ActiveX controls in their applications.

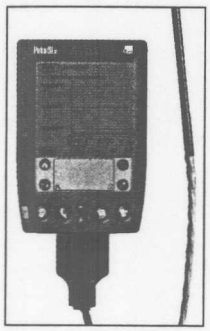


FIGURE 5 A PDA's serial port provides a convenient built-in means to communicate with an external device. The temperature-measurement shown here simply clips onto a Palm Pilot.
Courtesy of Tescina.

Run without Visual Basic

You don't have to use Visual Basic 6.0 to take advantage of Basic. The NS Basic/Palm 2.0 software provides another software-development tool that lets programmers write code using the Basic language. But unlike AppForge, NS Basic/Palm dispenses with the underlying framework of Visual Basic. Thus, if you don't have VB 6.0, you don't need to buy it. NS Basic/Palm comes in several versions, from the single-user package ($149) to enterprise versions (base price: $399).

The layout of the NS Basic/Palm IDE looks much like a cross between VB and a target running the Palm OS (**Figure 6**). You can move visual elements such as check boxes, labels, pop-up menus, fields, and other objects onto the target's "screen" and then connect the objects using Basic code. If you're at all familiar with Basic, you'll feel at home using NS Basic/Palm. The software provides standard commands, math functions (including trig operations), string manipulations, and decision-making operations, along with added built-in database functions and serial and infrared I/O operations. In all, the software provides more than 180 commands, including time/date and database-manipulation commands. You can set up databases, subroutines, data structures, and arrays within your programs.

You will need the POSE to test your software, assuming you don't want to test and debug on your target system. The POSE software comes on the NS Basic/Palm CD-ROM, but you'll have to download a ROM image from the PalmOS Web site (www.palmos.com) or from your target hardware. The compiled software produced by the NS Basic/Palm program requires that you store a supplied 88-kbyte run-time library, NSBRuntime.prc, in your target system. Even if you plan to run several applications, you need only one run-time library in your target. NS Basic requires no fees, licenses, or royalties for its run-time library, so you can distribute applications as you wish.

▶The Web version of this article contains additional information on math extensions, Palm OS Web resources and Palm OS programming tips.

You can elect to include the run-time library with your application code as a separate program, or you can compile it within your application. NS Basic calls the latter type of program a fat application, or "fat app." Placing several fat apps in a Palm OS target wastes memory by duplicating the run-time library for each application. In any case, the run-time software is transparent to other applications.

Send data back and forth

The application you develop using the tools mentioned above might acquire data that you want to transfer to a PC for analysis, storage, transmission on a network, or other use. Your application could use the Palm OS target's serial port or infrared port to communicate with the host PC. But a more convenient and elegant communication channel exists: a conduit.

Conduits employ Palm Computing's HotSync software that "synchronizes" the data in a Palm OS target with information saved on a PC. A HotSync program resides on a PC and runs when a user places a PDA in a docking cradle (supplied by the PDA manufacturer) and presses a button on the cradle. In many consumer applications, the HotSync software keeps an address book, e-mail messages, and calendar up to date by transferring the latest version to the PC or to the PDA while overwriting an older version.

The conduit software does not run on the target system, but remains within the

Commercial sources

PDA manufacturers

Handspring
Mt. View, CA
650-230-5000
www.handspring.com

Nokia
Irving, TX
972-894-5000
www.nokia.com

Palm Computing
Santa Clara, CA
408-878-900
www.palm.com

Sony Electronics
San Diego, CA
941-768-7669
www.sel.sony.com
www.us.sonypdadev.com/top.html

Symbol Technologies
Holtsville, NY
800-722-6234
www.symbol.com

Tescina
Fremont, CA
510-713-8001
www.tescina.com

Software companies

AppForge
Atlanta, GA
678-686-9000
www.appforge.com
AppForge offers demo versions of its software that you can download from the company's Web site. Look under "Free Downloads."

efig.com
Mountain View, CA
408-739-8002
www.efig.com

Metrowerks
Austin, TX
512-997-4700
www.metrowerks.com
Look under "development tools," and then under "standard platforms." You'll find the demo software under: www.metrowerks.com/contact/secure/palmdemo. Or you can use the site's search engine and the keyword "demo" to locate the proper page. You'll find the company's free tutorials at www.codewarrioru.com.

NS Basic Corp.
Toronto, ON, Canada
416-264-5999
www.nsbasic.com
NS Basic offers a demo version of the NS Basic/Palm software (1.7 Mbytes) on its Web home page.

Pacific Microinstruments
Sierra Madre, CA
626-836-6701
www.PacificMicroinstruments.com
PMI developed several types of measuring instruments that use the Handspring Springboard slot to connect with a Visor PDA. The company does custom development work.

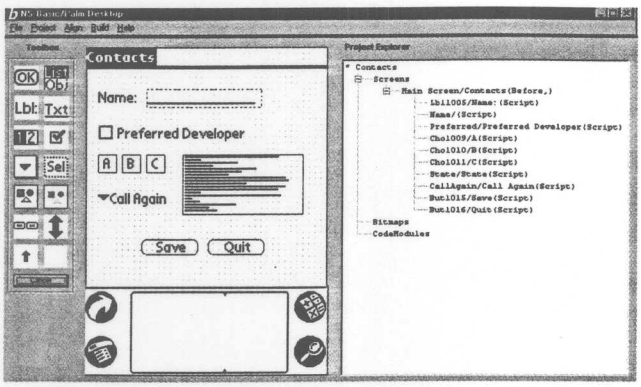


FIGURE 6 The screen for NS Basic/Palm looks much like the face of a Palm Pilot. Developers move buttons, text boxes, and other items to the Palm-like screen and "connect" them using Basic-language code. *Courtesy of NS Basic.*

host PC. You would use Visual C++, Visual Basic, or perhaps Java in conjunction with the Palm Conduit Development Kit (CDK) available from Palm to produce a special conduit. The CDK provides a generic conduit and the synchronization logic you can customize.

Conduits go beyond the scope of the programming tools described in this article, but you should know they exist. Information on the Palm OS Web site provides a wealth of background, reference, and technical information about conduits. Your best bet is to leave conduits to expert programmers or developers who have worked with them and can deliver a conduit suited to your needs.

PDAs that use the Palm OS can provide a convenient starting point for an instrument or instrument controller. The information in this article will give you a head start, and you can have a simple application running in a few days. But to take advantage of everything the Palm OS offers, you'll have to dig into manuals and information on Web sites. Don't expect too much of PDAs, though; no one designed them to be handheld instruments. But a PDA's limitations shouldn't stop you from considering it for a neat new application. **T&MW**

For more information

Bachmann, Glenn, *Palm Programming*, Sams Publishing, Indianapolis, IN, 1999.

Giguere, Eric, *Palm Database Programming: The Complete Developer's Guide*, John Wiley & Sons, New York, NY, 1999.

Rhodes, Neil, and Julie McKeehan, *Palm Programming: The Developer's Guide*, O'Reilly & Associates, Sebastopol, CA, 1998.

Jon Titus *has written real-time software and designed embedded systems and computer/instrument interfaces. He worked in electronics for 10 years and spent nine years at* EDN *magazine prior to joining* T&MW *in 1993. He has a BS from WPI, an MS from RPI, and a PhD from VPI. E-mail: jontitus@tmworld.com.*